

## Prime number

17  $\Rightarrow$  2, 3, 4, 5, ... 16  $\Rightarrow$  loop (2 to n-1)

↓  
not divisible by any  
↓  
Prime number

## Prime number function

```
class Solution {
public:
    int countPrimes(int n) {
        int count = 0;
        for(int i=2; i<n; i++){
            if(isPrime(i) == true){
                count++;
            }
        }
        return count;
    }
    public static boolean isPrime(int n){
        for(int i=2; i<n; i++){
            if(n%i==0){
                return false;
            }
        }
        return true;
    }
}
```

$\Rightarrow n^2 \Rightarrow$  Time

$\Rightarrow n$

TLE = ?

compiler  $\rightarrow$  1s  $\Rightarrow 10^8$  instructions

$$n = 10^6$$

$$n^2 = 10^6 \cdot 10^6 = 10^{12} \text{ instructions}$$

TLE

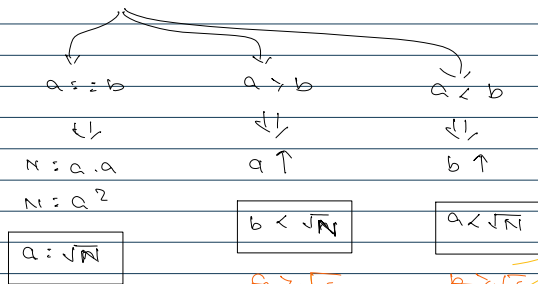
$$N = a \times b$$

$N \rightarrow$  prime  
 $a, b = 1, N$

$a, b \Rightarrow$  factors

$$N = a \times b$$

$N \rightarrow$  not prime



One of the factors will be found till  $\sqrt{n}$

$$81 \Rightarrow \sqrt{81} = 9$$

$$27 \times 3 = 81$$

$\rightarrow 1, 3, 9, \cancel{27}$

$$32 \Rightarrow 1, 2, 4, \cancel{8}, \cancel{16}, \cancel{32}$$

$$\sqrt{32} = 5 \text{ to } 6$$

$$i < \sqrt{n}$$

1)

$$i^2 < n$$

$$i * i < n$$

$\Rightarrow$  same inequalities



```

class Solution {
public int countPrimes(int n) {
    int count = 0;
    for(int i=2; i<n; i++){
        if(isPrime(i) == true){
            count++;
        }
    }
    return count;
}

public static boolean isPrime(int n){
    for(int i=2; i*i <= n; i++){
        if(n%i==0){
            return false;
        }
    }
    return true;
}
}

```

$$Time = n \cdot \sqrt{n}$$

$$n = 10^6$$

$$10^6 \cdot \sqrt{10^6}$$

$$10^6 \cdot (10^6)^{1/2}$$

$$10^6 \cdot 10^{6/2}$$

$$10^6 \cdot 10^3 = 10^9$$

TLE

## Sieve of Eratosthenes

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----

Prime numbers

2 3 5 7

11 13 17 19 23

29 31 37

41 43 47

53 59 67

71 73 79 83

89 97 101 103

107 109 113 127

131 137 149 151

157 163 167 173

179 181 187 191

193 197 199

$$\frac{n}{2} + \frac{n}{3} + \frac{n}{5} + \frac{n}{7}$$

$$\sqrt{100} = 10$$

$$n \left[ \frac{1}{2} + \frac{1}{3} + \frac{1}{5} + \frac{1}{7} + \dots \right]$$

$$n \left( \log_2(\log_2 N) \right)$$

$$\{1, O(n)\}$$

$$n \cdot \log_2(\log_2 N)$$

$$n = 10^6 = 2^{10}$$

$$\log_a b = b \cdot \log_a a$$

$$\log_a a = 1$$

$$2^{10} \cdot \log_2(2^{10})$$

$$2^{10} \cdot \log_2(10 \cdot \log_2 2^{10})$$

$$2^{10} \cdot \log_2 10$$

$$n \cdot 3.17 \sim n$$

## Palindromic Substring

### Variations:

1. Count all pal substrings

2. Find longest pal sub.

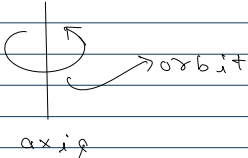
3. Count all odd-length pal sub.

4. even

5. Find the longest odd-length pal sub.

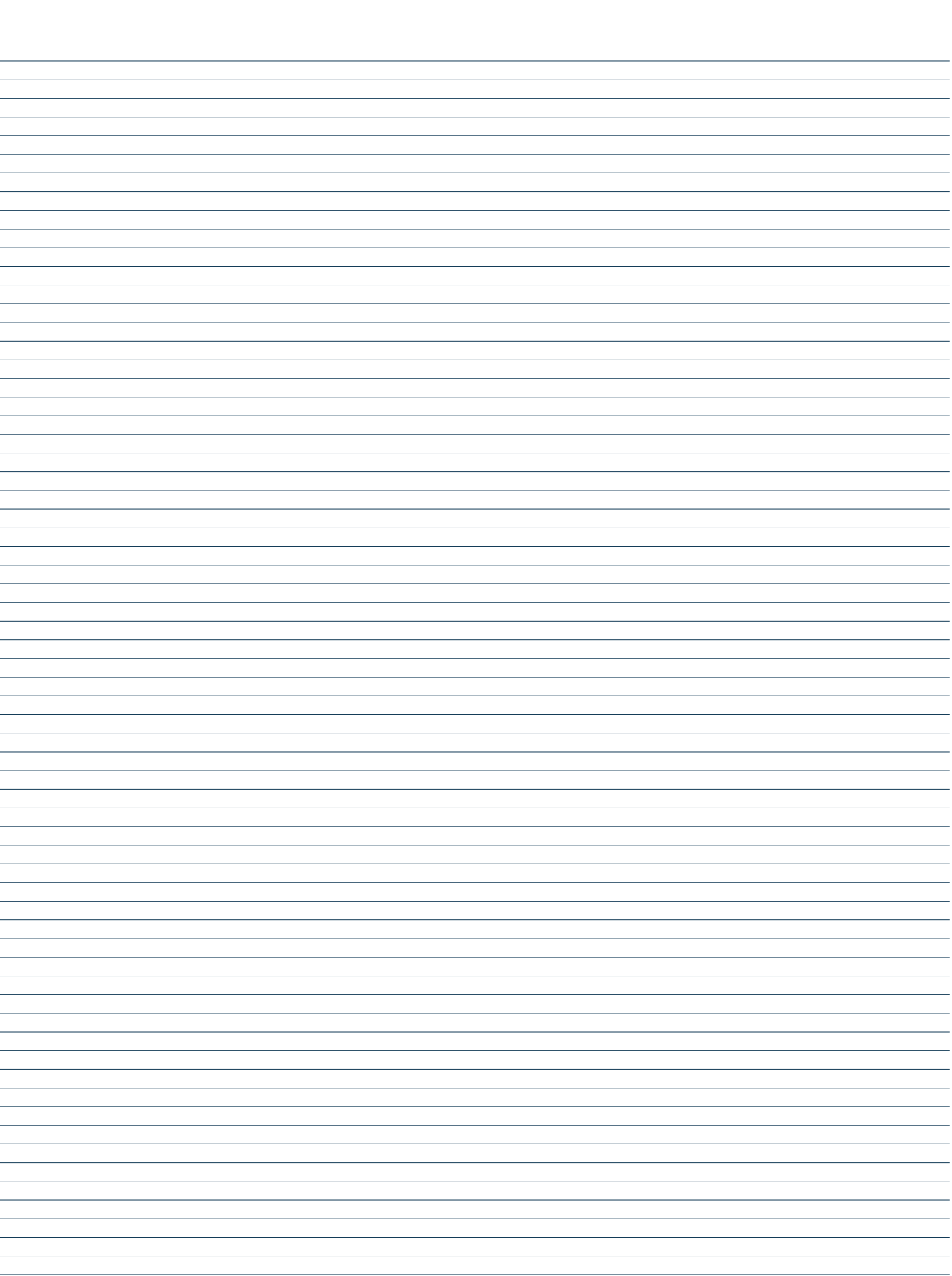
6. even

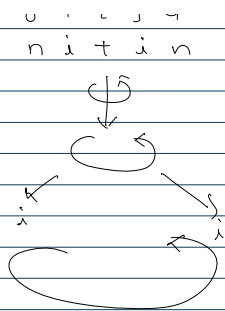
Axis Orbit Approach  
Expand around center approach



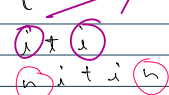
0 1 2 3 4  
n i t i n  
ψ

t  
i = i = pal  
1 1  
+ +





$t \rightarrow 1 = 1 \Rightarrow \text{pal}$



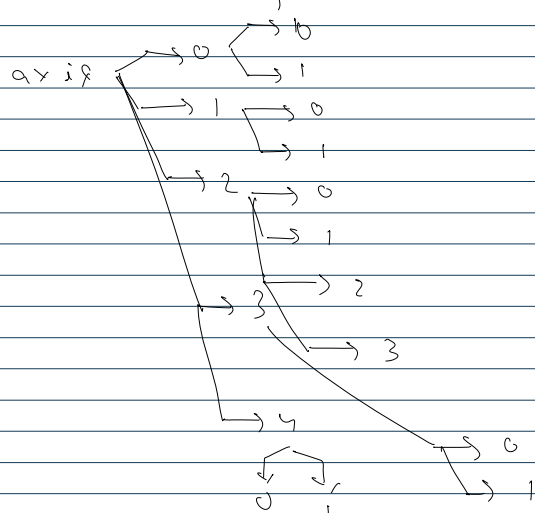
$n \rightarrow 4 \Rightarrow \text{pal}$

$t \Rightarrow \text{axis} = 2$   
 $i \Rightarrow \text{axis} = 2 \Rightarrow (1, 3)$   
 $n \Rightarrow \text{axis} = 2 \Rightarrow (0, 4)$   
 $\text{axis} - \text{orbit}$   
 $\text{axis} + \text{orbit}$

### odd length palindromes

<p>0 1 2 3 4 n i t i n</p> <p><math>\text{axis} = 0</math></p> <p><math>\text{orbit} = 0</math></p> <p> <math>0 - 0 = 0</math>  <math>0 + 0 = 0</math> </p> <p><math>0 - 1 = -1</math>  <math>0 + 1 = 1</math></p> <p><math>\text{orbit} = 1</math></p>	<p>0 1 2 3 4 n i t i n</p> <p><math>\text{axis} = 1</math></p> <p><math>\text{orbit} = 0</math></p> <p> <math>1 - 0 = 1</math>  <math>1 + 0 = 1</math> </p> <p><math>1 - 1 = 0</math>  <math>1 + 1 = 2</math></p> <p><math>\text{orbit} = 1</math></p>	<p>0 1 2 3 4 n i t i n</p> <p><math>\text{axis} = 2</math></p> <p><math>\text{orbit} = 0</math></p> <p> <math>2 - 0 = 2</math>  <math>2 + 0 = 2</math> </p> <p><math>2 - 1 = 1</math>  <math>2 + 1 = 3</math></p> <p><math>2 - 2 = 0</math>  <math>2 + 2 = 4</math></p> <p><math>\text{orbit} = 2</math></p>	<p>0 1 2 3 4 n i t i n</p> <p><math>\text{axis} = 3</math></p> <p><math>\text{orbit} = 0</math></p> <p> <math>3 - 0 = 3</math>  <math>3 + 0 = 3</math> </p> <p><math>3 - 1 = 2</math>  <math>3 + 1 = 4</math></p> <p><math>\text{orbit} = 1</math></p>	<p>0 1 2 3 4 n i t i n</p> <p><math>\text{axis} = 4</math></p> <p><math>\text{orbit} = 0</math></p> <p> <math>4 - 0 = 4</math>  <math>4 + 0 = 4</math> </p> <p><math>4 - 1 = 3</math>  <math>4 + 1 = 5</math></p> <p><math>\text{orbit} = 1</math></p>
---	--	--	--	--

axis - orbit  
axis + orbit



```

public class Main {
    public static void main(String[] args) {
        String s = "nitin";
        System.out.println(palindromicSubstringCount(s));
    }
}

```

```

public static int palindromicSubstringCount(String s){
    // Odd length
    int odd = 0;
    for(int axis=0; axis<s.length(); axis++){
        String temp = "";
    }
}

```

5

6

7

```

for(int orbit=0; axis-orbit >= 0 && axis+orbit < s.length(); orbit++){
    if(s.charAt(axis-orbit) == s.charAt(axis+orbit)){
        odd++;
        if(axis-orbit != axis+orbit){
            temp = s.charAt(axis-orbit) + temp + s.charAt(axis+orbit);
        }
    }
    else{
        temp = "" + s.charAt(axis-orbit);
    }
    System.out.println(temp);
}
}
return odd;
}
}

```

Even length substring

$$\begin{cases} \text{axis-orbit} & \text{axis} \rightarrow 0.5 \\ \text{axis+orbit} & \text{orbit} \rightarrow 0.5 \end{cases}$$

0 1 2 3  
n a a n  
axis=0.5

0 1 2 3  
n a a n  
axis=1.5

0 1 2 3  
n a a n  
axis=2.5

0 1 2 3  
n a a n  
axis=3.5

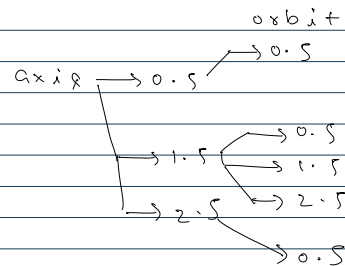
$0.5 - 0.5 = 0 \} n \neq a$   
 $0.5 + 0.5 = 1 \} X$

$1.5 - 0.5 = 1 \} a = a$   
 $1.5 + 0.5 = 2 \} aa$

$2.5 - 0.5 = 2 \} a \neq n$   
 $2.5 + 0.5 = 3 \} X$

$1.5 - 1.5 = 0 \} n = n$   
 $1.5 + 1.5 = 3 \} naan$

$1.5 - 2.5 = -1 \} X$   
 $1.5 + 2.5 = 4 \} X$



merge 2 sorted arrays

arr1  $\rightarrow$  2, 3, 5

arr2  $\rightarrow$  1, 3, 5, 7, 8

arr  $\rightarrow$  1, 2, 3, 3, 5, 6, 7, 8

2, 3, 5, 1, 3, 6, 7, 8

11  
2008  $\rightarrow$  Bubble  
 $\rightarrow n \cdot \log n$   
11  
 $\rightarrow$  merge

1, 2, 3, 3, 5, 6, 7, 8

arr1 complete

0 1 2  
2 3 5

0 1 2 3 4  
1 3 6 7 8

0 1 2 3 4 5 6 7  
1 2 3 3 5

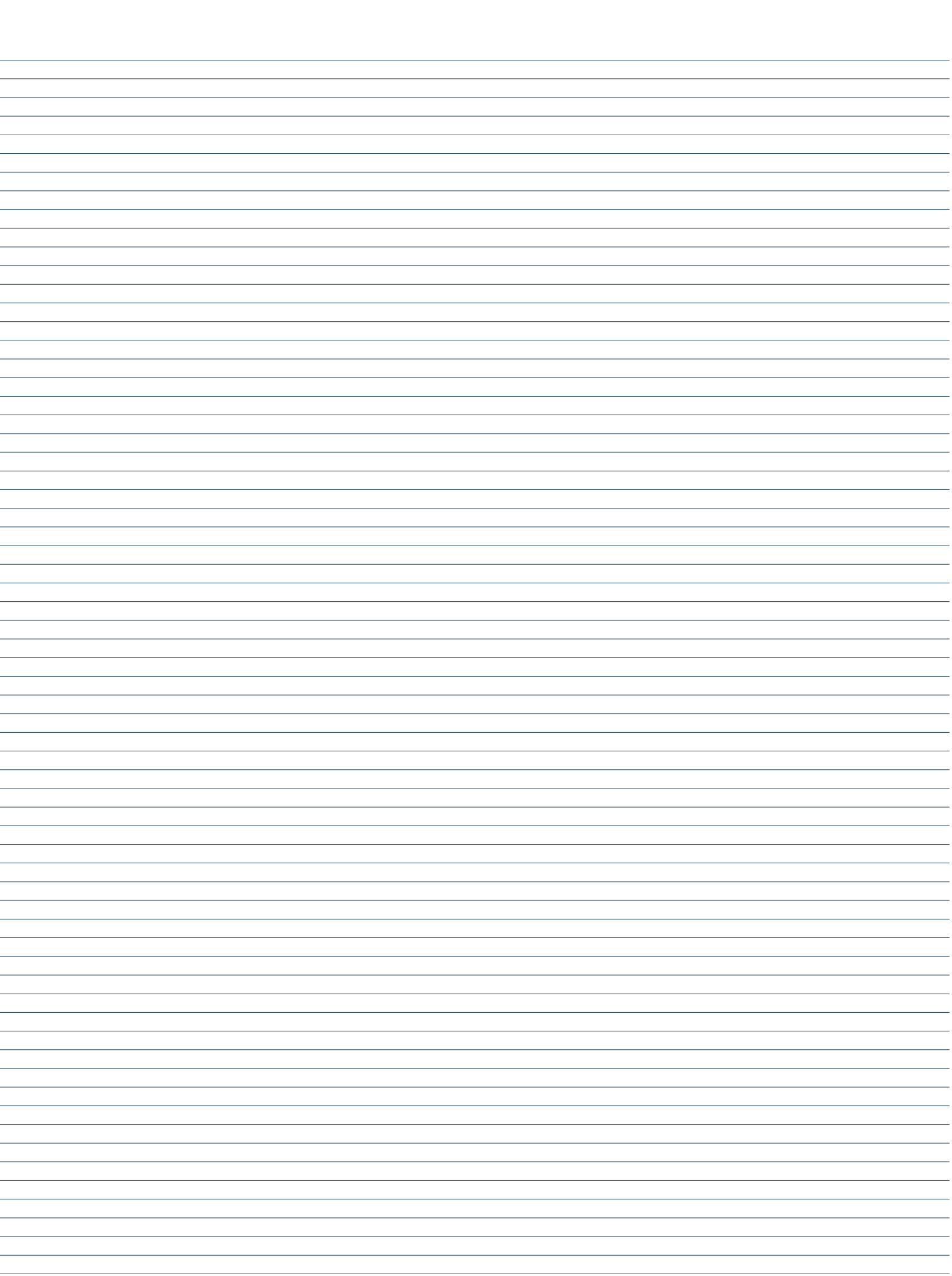
i: 0  
j: 0  
k: 0

$arr1[i] < arr2[j] \Rightarrow arr1[i] = arr[k], i++, k++$   
 $arr1[i] > arr2[j] \Rightarrow arr2[j] = arr[k], j++, k++$

```

public class Main {
    public static void main(String[] args) {

```





```
int[] arr1 = {2, 3, 5};  
int[] arr2 = {1, 3, 6, 7, 8};
```

```
int[] arr = merge(arr1, arr2);
```

```
for(int i=0; i<arr.length; i++){  
    System.out.print(arr[i] + " ");  
}
```

```
public static int[] merge(int[] arr1, int[] arr2){  
    int n = arr1.length;  
    int m = arr2.length;
```

```
    int[] arr = new int[n+m];
```

```
    int i = 0;  
    int j = 0;  
    int k = 0;
```

```
    while(i<n && j<m){  
        if(arr1[i] < arr2[j]){  
            arr[k] = arr1[i];  
            i++;  
            k++;  
        }  
        else{  
            arr[k] = arr2[j];  
            j++;  
            k++;  
        }  
    }  
}
```

```
    while(i<n){  
        arr[k] = arr1[i];  
        i++;  
        k++;  
    }
```

```
    while(j<m){  
        arr[k] = arr2[j];  
        j++;  
        k++;  
    }
```

```
    return arr;  
}
```

```
}
```

